

We claim:

1. A method comprising:

renaming a data file of a computer system while permitting lock-free look-ups to the data file by utilizing a temporary record to which look-ups to the data file resolve while
5 the data file is being renamed.

2. The method of claim 1, wherein renaming the data file comprises atomically renaming the data file.

3. The method of claim 2, wherein atomically renaming the data file of the computer file system while permitting lock-free look-ups to the data file comprises atomically
10 renaming the data file of the computer file system such that once a look-up of an old name of the data file necessarily fails, any subsequent look-up of a new name of the data file necessarily succeeds, and such that once a look-up of the new name of the data file necessarily succeeds, any subsequent look-up of the old name of the data file necessarily fails.

15 4. The method of claim 1, wherein renaming the data file of the computer file system while permitting lock-free look-ups to the data file comprises:

creating a temporary record within a hash chain encompassing a record for the data file and corresponding to a location of the data file within the computer file system;

linking the temporary record within the hash chain so that the temporary record points
20 to a before record to which the record for the data file points and to an after record to

which the record for the data file points;

renaming the record for the data file with a new name; and,

removing the temporary record from the hash chain,

wherein while the temporary record exists within the hash chain, look-ups of the data

5 file resolve to the temporary record, such that the temporary record causes the look-ups to wait until the record for the data file has been renamed and the temporary record removed.

5. The method of claim 1, further comprising looking up the data file in a lock-free manner while the data file is being atomically renamed.

10 6. The method of claim 5, wherein looking up the data file in the lock-free manner comprises, while traversing records of a containing construct of a record for the data file to locate the record for the data file:

where a number of the records of the containing construct traversed so far exceeds a predetermined maximum number of records, reverting to a locking process to look up the
15 data file;

where a currently traversed record of the containing construct has already been traversed, returning failure of the look-up of the data file;

where a currently traversed record of the containing construct does not match the data file, continuing traversal of the records of the containing construct;

20 where a currently traversed record of the containing construct matches the data file and is a temporary record for the data file, waiting until the temporary record has been removed and restarting look-up of the data file; and,

where a currently traversed record of the containing construct matches the data file and is the record for the data file, returning the record such that look-up of the data file has succeeded.

7. The method of claim 6, wherein the containing construct of the record for the data file
5 is one of a hash chain and a sub-directory.

8. A method comprising:

atomically moving a data file from one location to another location within a computer file system while permitting lock-free look-ups to the data file by utilizing a temporary record to which look-ups to the data file resolve while the data file is being moved.

10 9. The method of claim 8, wherein atomically moving the data file from one location to another location within the computer file system comprises atomically moving the data file from one location to another location within the computer file system such that once a look-up of the data file within an old location of the data file necessarily fails, any subsequent look-up of the data file within a new location of the data file necessarily
15 succeeds, and such that once a look-up of the data file within the new location necessarily succeeds, any subsequent look-up of the data file within the old location necessarily fails.

10. The method of claim 8, wherein atomically moving the data file from one location to another location within the computer file system comprises:

creating a temporary record within a hash chain encompassing a record for the data
20 file and corresponding to an old location of the data file within the computer file system;

linking the temporary record within the hash chain so that the temporary record points to a before record to which the record for the data file points and to an after record to which the record for the data file points;

moving the record for the data file to another hash chain corresponding to a new

5 location of the data file within the computer file system; and,

removing the temporary record from the hash chain,

wherein while the temporary record exists within the hash chain, look-ups of the data file resolve to the temporary record, such that the temporary record causes the look-ups to wait until the record for the data file has been moved to the other hash chain and the

10 temporary record removed.

11. The method of claim 8, further comprising, while traversing records of a containing construct of a record for the data file to locate the record for the data file:

where a number of the records of the containing construct traversed so far exceeds a predetermined maximum number of records, reverting to a locking process to look up the
15 data file;

where a currently traversed record of the containing construct has already been traversed, returning failure of the look-up of the data file;

where a currently traversed record of the containing construct does not match the data file, continuing traversal of the records of the containing construct;

20 where a currently traversed record of the containing construct matches the data file and is a temporary record for the data file, waiting until the temporary record has been removed and restarting look-up of the data file; and,

where a currently traversed record of the containing construct matches the data file

and is the record for the data file, returning the record such that look-up of the data file has succeeded.

12. A method comprising:

creating a temporary record within a hash chain encompassing a record for a data file

5 and corresponding to a location of the data file within a computer file system;

linking the temporary record within the hash chain so that the temporary record points to a before record to which the record for the data file points and to an after record to which the record for the data file points;

renaming the record for the data file with a new name; and,

10 removing the temporary record from the hash chain,

wherein while the temporary record exists within the hash chain, look-ups of the data file resolve to the temporary record, such that the temporary record causes the look-ups to wait until the record for the data file has been renamed and the temporary record has been removed.

15 13. The method of claim 12, further comprising, prior to removing the temporary record from the hash chain, moving the record for the data file to another hash chain corresponding to a new location of the data file within the computer file system, wherein while the temporary record exists within the hash chain, look-ups of the data file resolve to the temporary record, such that the temporary record causes the look-ups to wait until
20 the record for the data file has been renamed and moved to the other hash chain and the temporary record has been removed.

14. The method of claim 12, further comprising looking-up the data file in a lock-free manner, by traversing records of the hash chain encompassing the record for the data file to locate the record for the data file, and where a currently traversed record of the hash chain matches the data file and is a temporary record for the data file, waiting until the
5 temporary record has been removed and then restarting look-up of the data file

15. A method comprising:

creating a temporary record within a hash chain encompassing a record for a data file and corresponding to a location of the data file within a computer file system;

linking the temporary record within the hash chain so that the temporary record points
10 to a before record to which the record for the data file points and to an after record to which the record for the data file points;

moving the record for the data file to another hash chain corresponding to a new location of the data file within the computer file system; and,

removing the temporary record from the hash chain,

15 wherein while the temporary record exists within the hash chain, look-ups of the data file resolve to the temporary record, such that the temporary record causes the look-ups to wait until the record for the data file has been moved and the temporary record removed.

16. The method of claim 15, further comprising, prior to moving the record for the data file to the other hash chain, renaming the record for the data file with a new name,

20 wherein while the temporary record exists within the hash chain, look-ups of the data file resolve to the temporary record, such that the temporary record causes the look-ups to

wait until the record for the data file has been renamed and moved to the other hash chain and the temporary record removed.

17. The method of claim 15, further comprising looking-up the data file in a lock-free manner, by traversing records of the hash chain encompassing the record for the data file
5 to locate the record for the data file, and where a currently traversed record of the hash chain matches the data file and is a temporary record for the data file, waiting until the temporary record has been removed and then restarting look-up of the data file.

18. A system comprising:

a file system in which a plurality of files are stored; and,

10 means for at least one of atomically renaming and atomically moving from one location to another one of the files within the file system, while permitting lock-free file look-ups.

19. The system of claim 18, wherein the system comprises a plurality of computing nodes over which the file system is implemented.

15 20. The system of claim 18, wherein the system comprises a single computing node within which the file system is implemented.

21. An article of manufacture comprising:

a computer-readable medium; and,

means in the medium for atomically renaming and atomically moving from one

location to another within a file system a data file, while permitting lock-free file look-ups.

22. An article of manufacture comprising:

a computer-readable medium; and,

5 means in the medium for atomically renaming within a file system a data file, while permitting lock-free file look-ups.

23. An article of manufacture comprising:

a computer-readable medium; and,

10 means in the medium for atomically moving from one location to another within a file system a data file, while permitting lock-free file look-ups.